https://github.com/IoanaDudas/LFTC

Finite Automata

The FiniteAutomata is structured as a class with 5 fields: Q - states, E (Σ) - alphabet, q0 - initial state, F - final states, S - transitions.

The transitions S are kept in a *HashMap*, where each pair (source, route) - represented by a *SimpleEntry* is mapped to a *List* of destination states, for example: (A, 1) -> [A, B], meaning A goes to A or B with value 1.

DFA

Checking that the FA is a DFA is done by going through all the HashMap keys and looking if there is any list with a length greater than 1, then it returns *False* and if it cannot find any, *True*.

Sequence accepted

Checking if a sequence is accepted by the FA is done by going through each symbol of the given sequence and checking if that route exists, otherwise it returns *False*. If all the routes can be reached, at the end it checks if the state we reached is a final state and if not it returns *False*, or *True* if it is a final state.

Integration with Scanner

In the Scanner class, by changing the regex matching for numbers and identifiers with checking if the FA (which is read from different files for identifiers and constants) accepts a given sequence, namely one which represents an identifier or a number, the FA can be used to detect tokens.